

University of Groningen

Additive and multiplicative two-level spectral preconditioning for general linear systems

Carpentieri, Bruno; Giraud, Luc; Gratton, Serge

Published in:
SIAM: Journal on Scientific Computing

DOI:
[10.1137/060654906](https://doi.org/10.1137/060654906)

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Publisher's PDF, also known as Version of record

Publication date:
2007

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):

Carpentieri, B., Giraud, L., & Gratton, S. (2007). Additive and multiplicative two-level spectral preconditioning for general linear systems. *SIAM: Journal on Scientific Computing*, 29(4), 1593–1612. <https://doi.org/10.1137/060654906>

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

ADDITIVE AND MULTIPLICATIVE TWO-LEVEL SPECTRAL PRECONDITIONING FOR GENERAL LINEAR SYSTEMS*

B. CARPENTIERI[†], L. GIRAUD[‡], AND S. GRATTON[§]

Abstract. In this paper we introduce new preconditioning techniques for the solution of general symmetric and unsymmetric linear systems $Ax = b$. These approaches borrow some ideas of the multigrid philosophy designed for the solution of linear systems arising from the discretization of elliptic partial differential equations. We attempt to improve the convergence rate of a prescribed preconditioner M_1 . In a two-grid framework, this preconditioner is viewed as a smoother and the coarse space is spanned by the eigenvectors associated with the smallest eigenvalues of M_1A . We derive both additive and multiplicative variants of the resulting iterated two-level preconditioners for unsymmetric linear systems that can also be adapted for Hermitian positive definite problems. We show that these two-level preconditioners shift the smallest eigenvalues to one and tend to better cluster around one those eigenvalues that M_1 already succeeded in moving into the neighborhood of one. We illustrate the behavior of our method through extensive numerical experiments on a set of general linear systems. Finally, we show the effectiveness of these approaches on two challenging real applications; the first comes from a nonoverlapping domain decomposition method in semiconductor device modeling, the second from industrial electromagnetism applications.

Key words. iterative methods, Krylov methods, adaptive preconditioning, additive and multiplicative two-grid cycles, spectral preconditioner, deflation techniques, electromagnetic scattering applications, domain decomposition

AMS subject classifications. 65F10, 65F50, 65F15, 65R20, 65N38

DOI. 10.1137/060654906

1. Introduction. The large linear systems involved in many intensive numerical simulations are solved using preconditioned Krylov solvers. It is well known that the convergence of these linear solvers often depends to a large extent on the eigenvalue distribution of the preconditioned matrix. In many cases, it is observed that “removing” the extreme eigenvalues can greatly improve the convergence. Several techniques have been proposed in the past few years that attempt to tackle this problem. The proposed approaches can be split into two main families, depending on whether the scheme enlarges the generated Krylov space (see, for instance, [22, 25, 30, 33]) or adaptively updates the preconditioner [1, 6, 11, 20, 27, 29]. Many of these preconditioning techniques based on approximate eigenpairs have been proposed in the past few years. The underlying driving idea behind these approaches is to capture in a low-dimensional space the modes that do not quickly converge with a first-level preconditioner. In order to be efficient and keep the dimension of the low-dimensional space reasonably small, these techniques are generally used in combination with a first-level preconditioner that does a good job of clustering most eigenvalues close to one with relatively few outliers near the origin [10, 26, 36]. These spectral preconditioners can also be split into two main families, depending on their effect on the spectrum. They are referred to as deflation preconditioners [12, 14, 28] if they at-

*Received by the editors March 23, 2006; accepted for publication (in revised form) January 24, 2007; published electronically July 18, 2007. The major part of this work was carried out while the authors were working at CERFACS.

<http://www.siam.org/journals/sisc/29-4/65490.html>

[†]Karl-Franzens-University Graz, Institute of Mathematics and Scientific Computing, Heinrichstraße 36, A-8010 Graz, Austria (bruno.carpentieri@uni-graz.at).

[‡]ENSEEIH-IRIT, 2 Rue C. Camichel, 31 071 Toulouse Cedex, France (giraud@enseeiht.fr).

[§]CERFACS, 42 Avenue G. Coriolis, 31057 Toulouse Cedex, France (gratton@cerfacs.fr).

tempt to move a subset of eigenvalues to a positive quantity σ ; they are referred to as coarse grid preconditioners [6, 14] if they only attempt to shift the subset close to σ . The name of those latter techniques comes from domain decomposition and was first introduced in [5]. For this reason $\sigma = 1$ is often considered in practice.

On the other hand, for the solution of a linear system $Ax = b$ arising from the discretization of partial differential equations (PDE), the multigrid methods are among the fastest techniques. The core of the multigrid algorithms is a two-grid procedure that is applied recursively. A classical two-grid cycle can be briefly described as follows. On the fine grid a few iterations of a smoother are applied that attempt to reduce the high frequencies of the error (i.e., the components of the error in the space spanned by the vectors associated with the largest eigenvalues of A). The residual is then projected on the coarse grid where the low frequencies (i.e., the components associated with the smallest eigenvalues) can be captured and the coarse error equation is solved. The error on the coarse space is prolonged back to the fine grid to update the approximation computed by the presmoothing phase and a few more steps of the smoother are applied. Finally, if the new iterate is not accurate enough, the two-grid cycle is applied iteratively. In classical multigrid, the coarse space is not defined explicitly through the knowledge of the eigencomponents but by the selection of a space that is expected to capture them. The scheme presented above is a multiplicative algorithm [19], but additive variants [2, 13, 34] also exist.

In this work, we apply some underlying ideas from the multigrid approach to design preconditioners for the solution of general symmetric and unsymmetric linear systems (not necessarily coming from PDE applications). We attempt to improve a prescribed preconditioner M_1 , which is used to define the smoother involved in the two-grid scheme. In many situations such a preconditioner (that is application-dependent) is able to cluster most of the eigenvalues close to one but still leaves a few close to the origin; this is a fairly common situation for a wide range of problems and preconditioners. In that framework we define the coarse space by the span of the eigenvectors associated with the smallest eigenvalues of M_1A that are computed explicitly (i.e., the components of the error that are not efficiently damped by the smoother). We denote by V the matrix having the eigenvectors of M_1A as its columns. In that context, the prolongation operator is $P = V$, the restriction operator is denoted by R , and the matrix involved in the coarse grid error problem is defined by a Petrov–Galerkin formula $A_c = RAP$. The new approaches handle similar ingredients to those exploited in [6] but use them in a completely different way following another philosophy. This results in particular in a different spectral transformation for the preconditioned matrices. We mention that our multiplicative approach is a generalization of the generalized global basis (GGB) method [36]. For a particular choice of M_1 , that is, an algebraic multigrid V-cycle, and a particular choice of V , the eigenvectors associated with the largest eigenvalues in modulus, our multiplicative method coincides with GGB if only one iteration of the smoother is performed. In section 2, we describe the proposed preconditioners and establish their effect on the spectrum of the preconditioned matrices. In section 3, we illustrate the numerical effects related to each parameter defining the new preconditioners on a set of unsymmetric and symmetric positive definite (SPD) linear systems from the Matrix Market [4]. We demonstrate their computational efficiency on very challenging linear systems that arise in two-dimensional unstructured mixed finite element calculation for semiconductor device modeling as well as for the solution of dense linear systems arising from boundary element methods in industrial electromagnetic applications.

2. Two-level spectral preconditioning. We consider the solution of the linear system

$$(1) \quad Ax = b,$$

where A is an $n \times n$ nonsingular matrix, and x and b are vectors of size n . The linear system is solved using a preconditioned Krylov solver, and we denote by M_1 the left preconditioner, meaning that we solve

$$(2) \quad M_1 Ax = M_1 b.$$

Let $\{\lambda_1, \dots, \lambda_n\}$ be the set of eigenvalues of $M_1 A$ in any order where the multiple eigenvalues are repeated. Let the columns of V_k be the basis of a right invariant subspace of $M_1 A$ of dimension k . Suppose without loss of generality that $M_1 A V_k = V_k J_k$, where the eigenvalues of J_k are $\{\lambda_1, \dots, \lambda_k\}$. In this section we describe two procedures that attempt to improve the eigenvalue distribution of the preconditioned matrix. We formulate the algorithms in the framework of algebraic multilevel methods; the terminology and the design principles are inherited from classical two-grid algorithms. In the next two sections we present these two variants of the algorithm that are either multiplicative or additive.

2.1. Multiplicative two-grid spectral preconditioning. A generic two-level multigrid cycle is illustrated in Algorithm 1, where M_1 is used to define a weighted stationary method that implements the smoother. The algorithm takes as input a vector r that is the residual vector we want to precondition and returns as output the preconditioned residual vector z . After μ_1 smoothing steps we project, using the restriction $R = W^H$, the residual into the coarse subspace and solve the coarse space error equation involving $A_c = W^H A V_k$. Finally, we prolongate back the error using V_k in the original space and smooth again the new approximation. The preconditioner constructed using this scheme depends on A , M_1 , μ_1 , μ_2 , ε , and $\omega > 0$ and could be denoted by $M_{Mul}(A, M_1, k, \omega, \mu_1, \mu_2)$. For the sake of simplicity of exposure, when no confusion is possible we will simply denote it by M_{Mul} or $M_{Mul}(A, M_1)$.

PROPOSITION 1. *Let W be such that $A_c = W^H A V_k$ has full rank; the preconditioning operation described in Algorithm 1 can be written in the form $z = M_{Mul} r$. For the case $iter = 1$ the preconditioner M_{Mul} has the following expression:*

$$(3) \quad M_{Mul} = A^{-1} - (I - \omega M_1 A)^{\mu_2} (I - M_c A) (I - \omega M_1 A)^{\mu_1} A^{-1},$$

where $M_c = V_k (W^H A V_k)^{-1} W^H$.

Proof. This is a fairly standard result; for its proof we refer, for instance, to [8]. \square

The following proposition describes the eigenvalue distribution of the preconditioned matrix $M_{Mul} A$. Similarly to multigrid, if we choose V_k to be the invariant space associated with the smallest eigenvalues (i.e., the closest to the origin), this result shows that the smallest eigenvalues are shifted to one and that those that were already close to one are clustered closer to this point. On the assumption that the initial preconditioner M_1 has done a good job of clustering most eigenvalues close to one with relatively few outliers near the origin, the effect of M_{Mul} is expected to be very beneficial.

ALGORITHM 1. *Multiplicative spectral preconditioner.*

```

1: set  $z^1 = 0$ 
2: for  $\ell = 1, iter$  do
3:   % Presmoothing: damp the high frequencies of the error %
4:    $s_1^0 = z^\ell$ 
5:   for  $j = 1, \mu_1$  do
6:      $s_1^j = s_1^{j-1} + \omega M_1(r - A s_1^{j-1})$ 
7:   end for
8:   % Coarse grid correction %
9:    $z_2^\ell = z_1^\ell + V_k A_c^{-1} W^H(r - A z_1^\ell)$ 
10:  % Postsmoothing: damp again the high frequencies of the error %
11:   $s_2^0 = z_2^\ell$ 
12:  for  $j = 1, \mu_2$  do
13:     $s_2^j = s_2^{j-1} + \omega M_1(r - A s_2^{j-1})$ 
14:  end for
15: end for
16:  $z = z^{iter}$ 

```

PROPOSITION 2. *The preconditioner M_{Mul} defined by Proposition 1 is such that the preconditioned matrix $M_{Mul}A$ has eigenvalues*

$$\begin{cases} \eta_i = 1 & \text{if } i < k, \\ \eta_i = 1 - (1 - \omega \lambda_i)^{\mu_1 + \mu_2} & \text{if } i \geq k. \end{cases}$$

Proof. We will show that the preconditioned matrix is similar to a matrix whose eigenvalues are those indicated. Let V_k^\perp be an orthogonal complement of V_k . We have

$$M_1 A [V_k V_k^\perp] = [V_k V_k^\perp] \begin{pmatrix} J_k & E \\ 0 & F \end{pmatrix},$$

where the eigenvalues of J_k are $\{\lambda_1, \dots, \lambda_k\}$ and those of F are $\{\lambda_{k+1}, \dots, \lambda_n\}$.

Thus $M_1 A V_k = V_k J_k$ and $M_c A V_k = V_k$. In addition, it is easy to show by induction on j that $(I - \omega M_1 A)^j V_k = V_k (I - \omega J_k)^j$ and $(I - \omega M_1 A)^j V_k^\perp = V_k^\perp (I - \omega F)^j + V_k P_j$, where $P_j \in \mathbb{R}^{k \times (n-k)}$. We also have $(I - M_c A) V_k^\perp = V_k^\perp - V_k P_c$ with $P_c \in \mathbb{R}^{k \times (n-k)}$. Then the following relations hold:

$$(4) \quad M_{Mul} A [V_k V_k^\perp] = [V_k V_k^\perp] \begin{pmatrix} I & * \\ 0 & I - (I - \omega F)^{\mu_1 + \mu_2} \end{pmatrix},$$

where the asterisk denotes a nonzero block whose actual expression is unimportant. The equality (4) concludes the proof. \square

Remark 1. It should be noted that the spectrum of the preconditioned matrix does not depend on the selection of W^H , which would play the role of the restriction operator in the multiplicative two-grid method.

2.2. Additive two-grid spectral preconditioning. In the additive algorithm, the coarse grid correction and the smoothing operation are decoupled. Each process generates an approximation of the preconditioned residual vector in complementary subspaces. The coarse grid correction computes only components in the space spanned by the eigenvectors associated with the few selected small eigenvalues, while at the end of the smoothing step the preconditioned residual is filtered so that only

components in the complementary subspace are retained. These two contributions are summed together for the solution update. A simple additive two-level multigrid cycle is illustrated in Algorithm 2. In this algorithm we follow [13] and select the procedure advocated in [34]: we define the filtering operators using the grid transfer operators as $(I - V_k W^H)$. This operator is supposed to remove all the components in the V_k directions. A natural choice is to select W^H so that $(I - V_k W^H)V_k = 0$ (i.e., $W^H V_k = I$).

ALGORITHM 2. *Additive spectral preconditioner.*

```

1: set  $z^1 = 0$ 
2: for  $\ell = 1, \text{iter}$  do
3:   % Compute the residual %
4:    $s^\ell = r - Az^\ell$ 
5:   % Compute the high and low frequency corrections %
6:   % 1- High frequency correction: Damp all the frequencies of the error and filter
   %
7:    $e_1^{\ell,0} = 0$ 
8:   for  $j = 1, \mu_1 + \mu_2$  do
9:      $e_1^{\ell,j} = e_1^{\ell,j-1} + \omega M_1(s^\ell - Ae_1^{\ell,j-1})$ 
10:  end for
11:  % Filter the high frequencies of the correction %
12:   $c_1^\ell = (I - V_k W^H)e_1^{\ell,\mu_1+\mu_2}$ 
13:  % 2- Low frequency correction %
14:   $c_2^\ell = V_k A_c^{-1} W^H s^\ell$ 
15:  % Update the solution %
16:   $z^{\ell+1} = z^\ell + c_1^\ell + c_2^\ell$ 
17: end for
18:  $z = z^{\text{iter}}$ 

```

PROPOSITION 3. Let W be such that $A_c = W^H A V_k$ has full rank and satisfies $(I - V_k W^H)V_k = 0$; the preconditioning operation described in Algorithm 2 can be written in the form $z = M_{\text{Add}} r$. In the case $\text{iter} = 1$ the preconditioner M_{Add} has the following expression:

$$(5) \quad M_{\text{Add}} = V_k A_c^{-1} W^H + (I - V_k W^H)(I - (I - \omega M_1 A)^{\mu_1 + \mu_2})A^{-1}.$$

Proof. The proof follows arguments similar to those for the proof of Proposition 1. \square

PROPOSITION 4. The preconditioner M_{Add} defined by Proposition 3 is such that the preconditioned matrix $M_{\text{Add}}A$ has eigenvalues

$$\begin{cases} \eta_i = 1 & \text{if } i \leq k, \\ \eta_i = 1 - (1 - \omega \lambda_i)^{\mu_1 + \mu_2} & \text{if } i > k. \end{cases}$$

Proof. Using arguments similar to those for the proof of Proposition 2, it can be shown that

$$(6) \quad M_{\text{Add}}A [V_k V_k^\perp] = [V_k V_k^\perp] \begin{pmatrix} I & * \\ 0 & I - (I - \omega F)^{\mu_1 + \mu_2} \end{pmatrix},$$

where the asterisk denotes a nonzero block whose actual expression is unimportant. The equality (6) concludes the proof. \square

2.3. The Hermitian positive definite linear systems. For Hermitian positive definite (HPD) linear systems a desirable property for M_1 is to be Hermitian positive definite; we denote by $M_1 = L_1 L_1^H$ its Cholesky decomposition. We note that in many situations the preconditioner M_1 is given in this factorized form as in the incomplete factorization [24], approximate inverse (AINV) [3], or factorized sparse approximate inverse (FSAI) [21]. In the rest of this section we review situations where M_{Mul} and M_{Add} are HPD.

PROPOSITION 5. *The preconditioner M_{Mul} with $W = V_k$ is HPD if $\mu_1 + \mu_2$ is odd.*

Proof. We do not give the details of the calculation, but using simple matrix manipulations it can be shown that

$$(7) \quad M_{Mul}(A, M_1) = L_1 M_{Mul}(L_1^H A L_1, I) L_1^H.$$

We first show that $M_{Mul}(L_1^H A L_1, I)$ is Hermitian. Because $L_1^H A L_1$ is HPD we have $L_1^H A L_1 = \tilde{V} D \tilde{V}^H$, where $D = \text{diag}(\lambda_1, \dots, \lambda_n)$ and $\tilde{V} = [\tilde{V}_k \tilde{V}_k^\perp]$. If we denote $D_k = \text{diag}(\lambda_1, \dots, \lambda_k)$ and $D_{\bar{k}} = \text{diag}(\lambda_{k+1}, \dots, \lambda_n)$, we have

$$(8) \quad M_{Mul}(L_1^H A L_1, I) = \tilde{V} \begin{pmatrix} D_k^{-1} & 0 \\ 0 & D_{\bar{k}}^{-1}(I - (I - \omega D_{\bar{k}})^{\mu_1 + \mu_2}) \end{pmatrix} \tilde{V}^H = \tilde{V} \hat{D} \tilde{V}^H.$$

Because $\mu_1 + \mu_2$ is odd, all diagonal entries of \hat{D} are positive (the function $f(x) = 1 - (1 - \omega x)^k$ is positive $\forall x \in (0, \infty)$ for k odd; see also Figure 1(b)). Furthermore, (7) leads to $M_{Mul} = (L_1 \tilde{V}) \hat{D} (L_1 \tilde{V})^H$. The Sylvester law of inertia implies that all the eigenvalues of M_{Mul} are positive. \square

PROPOSITION 6. *If $\mu_1 + \mu_2$ is even, the preconditioner M_{Mul} with $W = V_k$ is HPD if $\omega < 2\lambda_{\max}^{-1}(M_1 A)$.*

Proof. The proof is similar to that of Proposition 5. \square

In the HPD situation we have the following properties.

PROPOSITION 7. *Because $M_1 = L_1 L_1^H$, we can solve (1) using preconditioned conjugate gradient (PCG) with $M_{Mul}(A, M_1)$ as preconditioner. Another possibility is to solve $L_1^H A L_1 y = L_1^H b$ using PCG with preconditioner $M_{Mul}(L_1^H A L_1, I)$ and to recover $x = L_1 y$. Let x_k denote the iterate generated by the former approach, y_k the iterate associated with the second approach, and assume that the governing parameters of M_{Mul} are set so that it is HPD (see Proposition 5 or 6). If $x_0 = L_1 y_0$, then for any k , $x_k = L_1 y_k$.*

Proof. Because the preconditioners are HPD we consider their Cholesky decompositions, which are $M_{Mul}(A, M_1) = L_M L_M^H$ and $M_{Mul}(L_1^H A L_1, I) = \tilde{L}_M \tilde{L}_M^H$. From (7), we have $L_M L_M^T = L_1 \tilde{L}_M \tilde{L}_M^H L_1^H$, which shows that $L_M = L_1 \tilde{L}_M$ (uniqueness of the Cholesky decomposition). We now use the result on CG iterates from [17, 32] that states that solving $By = g$ with CG preconditioned by LL^T generates iterates y_k that can also be generated by unpreconditioned CG on $L^T B L \tilde{y} = L^T g$, provided that the initial guesses are such that $y_0 = L \tilde{y}_0$. We apply this result in the following sequence: CG on A preconditioned by $L_M L_M^T$ generates the same iterates as unpreconditioned CG on $L_M^T A L_M = \tilde{L}_M^H L_1^H A L_1 \tilde{L}_M$; these iterates are the same as those generated by CG on $L_1^H A L_1$ preconditioned by $\tilde{L}_M \tilde{L}_M^H$. \square

PROPOSITION 8. *In the HPD case, if $M_1 = I$ and $W = V_k$ and V_k is a set of orthonormal eigenvectors, then $M_{Mul} = M_{Add}$.*

Proof. Because A is HPD, A is diagonalizable in an orthonormal basis of eigenvectors. We then have that the vectors $W = V_k$ comply with the assumption $W^H V_k = I$ of Proposition 3.

With the assumptions that V_k is a set of eigenvectors, we have $M_{Mul}AV_k = V_k = M_{Add}AV_k$ and $M_{Mul}AV_k^\perp = M_{Add}AV_k^\perp$. \square

Remark 2. A situation where the assumptions of Proposition 8 hold is, for instance, when either $M_{Mul}(L_1^H AL_1, I)$ or $M_{Add}(L_1^H AL_1, I)$ is considered (i.e., the two-grid preconditioners are applied to $L_1^H AL_1 y = L_1^H b$). The additive preconditioner is in general not HPD even when A and M_1 are. One way to still use this preconditioner for an M_1 given in a symmetric factorized form $M_1 = L_1 L_1^H$ is to consider $M_{Add}(L_1^H AL_1, I)$ applied to $L_1^H AL_1 y = L_1^H b$. In that case we have $W = V_k$ and Proposition 8 shows that this preconditioner is the same as $M_{Mul}(L_1^H AL_1, I)$. Furthermore, Proposition 7 shows that CG then generates the same approximation with M_{Add} and M_{Mul} . This shows that for HPD linear systems all of these preconditioners are pretty much the same, and the choice of which to use can be made using floating-point arithmetic complexity as described below.

2.4. Some computational considerations. The spectral properties of M_{Mul} and M_{Add} have been derived from Algorithms 1 and 2, where only one outer iteration is considered. With calculations similar to those used so far, it can be shown that implementing “*iter*” outer steps leads to a spectral transformation for the two preconditioners that is defined by the following proposition.

PROPOSITION 9. *The preconditioner M_{Mul} defined by Algorithm 1 with $iter \geq 1$ is such that the preconditioned matrix $M_{Mul}A$ has eigenvalues*

$$\begin{cases} \eta_i = 1 & \text{if } i \leq k, \\ \eta_i = 1 - (1 - \omega \lambda_i)^{iter \times (\mu_1 + \mu_2)} & \text{if } i > k. \end{cases}$$

Similar proposition holds with M_{Add} defined by Algorithm 2.

From a computational point of view, applying M_{Mul} or M_{Add} does not require the same computational effort. In terms of application of M_1 both need $iter \times (\mu_1 + \mu_2)$ matrix-vector products. The number of matrix-vector products involving A is one less for M_{Add} than for M_{Mul} per cycle. For M_{Add} the number of products by A is

$$(9) \quad (\mu_1 + \mu_2 - 1) + (iter - 1) \times (\mu_1 + \mu_2).$$

Under the assumption that the convergence of the Krylov solver is mainly governed by the eigenvalue distributions (i.e., $iter \times (\mu_1 + \mu_2)$ given), Proposition 9 and the number of matrix-vector products given by (9) indicate that for M_{Mul} it is computationally more efficient to use only one two-grid cycle with $(iter \times (\mu_1 + \mu_2))$ smoothing steps rather than $iter$ two-grid cycles with $(\mu_1 + \mu_2)$ smoothing steps per cycle. For M_{Add} , the two variants have the same computational cost.

Remark 3. In [6], the initial linear system is never considered and the idea is to perform an appropriate low rank correction of the preconditioned matrix to shift part of its spectrum. Here, similarly to the two-grid schemes for PDE solution, we solve exactly the error equation of the unpreconditioned system in the invariant subspace associated with the smallest eigenvalues in modulus. In that respect, even though these approaches exploit the same ingredients, the underlying philosophies are different.

3. Numerical experiments. In this section we illustrate the numerical behavior of the two variants of the spectral algorithm described in the earlier section. In section 3.1 we investigate only the numerical behavior of the proposed schemes and do not address their computational relevance. Their computational efficiency strongly

depends on the relative costs of the smoother, the matrix-vector, and eigenmode calculation, which are application-dependent. Furthermore, the extra cost associated with the setup to compute the eigenvectors might be significant and might not be amortized by the savings in time for the linear system solution. Consequently these computational aspects are investigated on two challenging applications where the preconditioners have been implemented in the real simulation codes and run on front-end computers.

3.1. General test problems. We consider both unsymmetric and SPD systems. The numerical experiments are performed with MATLAB version 7.0. For the sake of simplicity we consider the $ILU(t)$ preconditioner [31] from MATLAB for M_1 in the unsymmetric case and the incomplete Cholesky factorization $IC(t)$ for SPD systems [24]. We report on results using the GMRES and the Bi-CGStab [35] solvers for unsymmetric problems, and the CG method for SPD systems. The experiments are carried out using left preconditioning for the unsymmetric problems. The initial guess for the solution is the zero vector. Because we compare different preconditioners, we choose as a stopping criterion the reduction of the normalized unpreconditioned residual by 10^{-6} , so that the stopping criterion is independent of the preconditioner. We explicitly compute the true unpreconditioned residual at each iteration. For the experiments with M_{Add} we use $W = QR^{-1}$, where $V_k = QR$ to ensure that $W^H V_k = I$. For the experiments with M_{Mul} we consider $W = V_k$. We mention that with these choices of W we have never encountered rank deficiency for A_c . However, it is reasonable to check the nonsingularity of A_c in practice to ensure that the preconditioners are defined; notice that in the HPD situation A_c cannot be singular. The eigenvectors V_k are computed in a preprocessing phase using the MATLAB function `eigs`.

In a first stage we consider both SPD and unsymmetric test matrices from the Matrix Market. In this section, we only report a few results representative of the general trends. For a complete description of the numerical experiments we refer the reader to the appendix of [8], where results on intensive experiments are reported.

For SPD matrices, we display the eigenvalue transformation operated by the preconditioners. These are the curves $f(x) = 1 - (1 - \omega x)^k \forall x > 0$ as the eigenvalues of $M_1 A$ are real positive. Figure 1 illustrates that the preconditioned matrices remain SPD for $(\mu_1 + \mu_2)$ odd, while ω should be set such that $\omega < 2\lambda_{max}^{-1}(M_1 A)$ for $(\mu_1 + \mu_2)$ even.

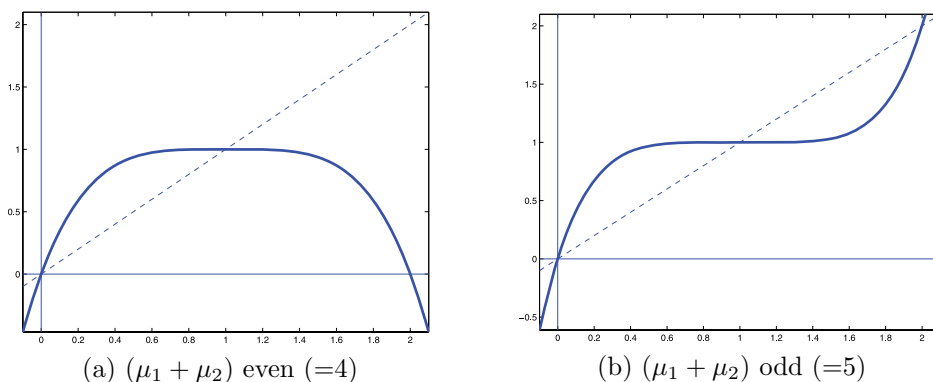


FIG. 1. Shape of the polynomial that governs the real eigenvalues distribution of the two-grid spectral preconditioner for $\omega = 1$.

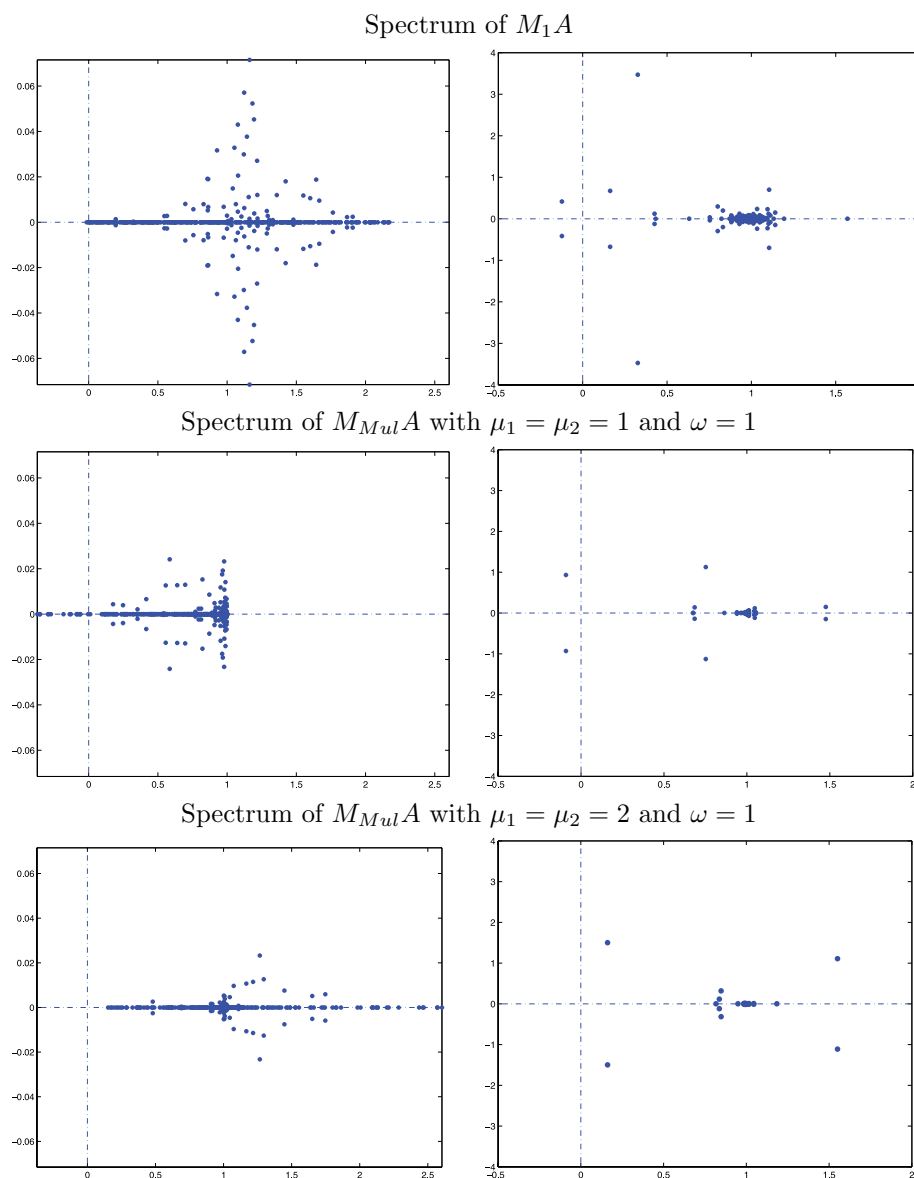


FIG. 2. Spectrum of the preconditioned matrices (BFW398A, left; GRE1107, right).

In Figure 2 we depict the spectrum of preconditioned matrices using only the first-level preconditioner and the two-grid preconditioner with different choices for μ_1 and μ_2 . In this figure, it can be seen that the spectral preconditioners do a good job in clustering close to one most of the eigenvalues. More precisely for $\mu_1 + \mu_2 > 1$ all the eigenvalues lying in the open disk of radius one centered in $(1, 0)$ are contracted toward $(1, 0)$; the ones out of this disk are moved away from $(1, 0)$. When $\mu_1 + \mu_2$ is odd, those that are real and larger than 2 become negative, whereas for $\mu_1 + \mu_2$ even they remain positive. This is illustrated by the example of the matrix BFW398A.

In the following subsections we consider the numerical behavior of the proposed

TABLE 1
Number of iterations with M_{Mul} and M_{Add} .

SAYLR1: $t = 3 \cdot 10^{-1}$; $\mu_1 = 2, \mu_2 = 1$; $\omega = 1.0$													
		Dimension of the coarse space											
		M_1	0	1	2	3	4	5	6	7	8	9	10
Bi-CGStab	M_{Add}	150	77	64	54	43	27	21	19	18	16	14	14
	M_{Mul}	150	77	63	49	40	28	22	19	17	19	13	14
GMRES(50)	M_{Add}	†	†	388	142	56	47	43	36	36	32	29	26
	M_{Mul}	†	†	441	141	56	47	40	39	36	33	29	26
GMRES(∞)	M_{Add}	116	72	63	56	49	43	36	33	33	30	26	23
	M_{Mul}	116	72	63	56	49	42	36	35	33	30	26	23

BFW398a: $t = 5 \cdot 10^{-1}$; $\mu_1 = 2, \mu_2 = 1$; $\omega = 1.0$													
		Dimension of the coarse space											
		M_1	0	1	2	3	4	5	6	7	8	9	10
Bi-CGStab	M_{Add}	112	73	48	44	33	28	23	23	16	14	14	14
	M_{Mul}	112	73	50	43	33	29	23	23	15	14	14	14
GMRES(50)	M_{Add}	286	106	80	52	45	40	34	30	27	25	23	22
	M_{Mul}	286	106	79	52	45	40	34	30	27	25	23	22
GMRES(∞)	M_{Add}	106	66	57	51	45	40	34	30	27	25	23	22
	M_{Mul}	106	66	57	51	45	40	34	30	27	25	23	22

preconditioners. The efficiency in terms of computational cost is very problem-dependent and is addressed in section 3.2, where two real-world applications are presented.

3.1.1. Numerical behavior of M_{Mul} versus M_{Add} . The two preconditioners M_{Add} and M_{Mul} give rise to preconditioned systems that have the same eigenvalue distribution but possibly different eigenspaces. In Table 1 we report on the number of iterations of the Krylov solvers for different matrices and different values of μ_1 and μ_2 when the size of the coarse space is varied. The symbol “ M_1 ” for the dimension of the coarse space means that no coarse space correction is needed and only one smoothing is applied; in that case the preconditioner reduces to the standard incomplete factorization. The symbol “†” indicates that the convergence was not observed in less than 1000 iterations. On all the numerical experiments we have performed, M_{Add} and M_{Mul} exhibit a very similar numerical behavior, as illustrated in Table 1. Small differences appear with Bi-CGStab, diminish with restarted GMRES, and vanish with full GMRES. However, from a numerical point of view none of the two preconditioners appears superior to the other. When the size of the coarse space is increased it can be seen that the two-level preconditioners monotonically improve the convergence of full GMRES. This is no longer strictly true for restarted GMRES and Bi-CGStab, but the trend is still to observe fewer iterations when the size of the coarse space is increased. Generally the larger the coarse space, the faster the convergence. One can see on the SAYLR1 matrix that even without coarse space correction few steps of smoothers can significantly improve the convergence of the Krylov solvers. On this latter example, the combination of the smoothing steps and the coarse grid correction is the only way to ensure the convergence of GMRES(50).

Because M_{Add} and M_{Mul} have a similar behavior and because M_{Mul} is naturally defined for SPD problems, we consider only M_{Mul} for the numerical experiments reported in the next three sections.

3.1.2. Effect of the number of smoothing steps. In Table 2 we display the number of iterations when the number of smoothing steps is varied from 1 to 3. It can

TABLE 2
 Number of iterations with M_{Mul} when the number of smoothing steps is varied.

S1RMQ4M1: $t = 1 \cdot 10^{-1}$; $\omega = \lambda_{\max}^{-1}(M_1 A)$													
		Dimension of the coarse space											
		M_1	0	1	2	3	4	5	6	7	8	9	10
CG	$\mu_1 + \mu_2 = 1$	204	204	134	114	104	90	89	89	83	82	78	77
	$\mu_1 + \mu_2 = 2$	204	150	97	83	76	65	64	64	60	60	56	56
	$\mu_1 + \mu_2 = 3$	204	123	80	68	62	53	53	53	49	49	46	46
BWM2000: $t = 3 \cdot 10^{-1}$; $\omega = 1.0$													
		Dimension of the coarse space											
		M_1	0	1	2	3	4	5	6	7	8	9	10
GMRES(20)	$\mu_1 + \mu_2 = 1$	66	66	56	36	19	16	12	11	11	11	11	10
	$\mu_1 + \mu_2 = 2$	66	18	16	13	11	9	8	7	7	7	7	7
	$\mu_1 + \mu_2 = 3$	66	15	13	11	9	7	6	6	6	6	6	6
FS-541-4: $t = 8 \cdot 10^{-1}$; $\omega = 1.0$													
		Dimension of the coarse space											
		M_1	0	1	2	3	4	5	6	7	8	9	10
GMRES(100)	$\mu_1 + \mu_2 = 1$	126	126	125	126	124	125	125	99	99	99	99	99
	$\mu_1 + \mu_2 = 2$	126	†	†	†	†	†	†	†	†	†	†	†
	$\mu_1 + \mu_2 = 3$	126	87	86	86	84	84	84	72	72	72	72	72
GMRES(∞)	$\mu_1 + \mu_2 = 1$	106	106	103	106	102	103	103	99	99	99	99	99
	$\mu_1 + \mu_2 = 2$	106	168	163	163	163	163	163	167	167	175	167	167
	$\mu_1 + \mu_2 = 3$	106	78	77	77	77	77	77	71	71	71	70	69
GRE1107: $t = 1 \cdot 10^{-2}$; $\omega = 1.0$													
		Dimension of the coarse space											
		M_1	0	1	2	3	4	5	6	7	8	9	10
GMRES(40)	$\mu_1 + \mu_2 = 1$	†	†	80	40	37	36	32	31	27	26	24	23
	$\mu_1 + \mu_2 = 2$	†	34	31	28	25	24	22	21	19	18	17	16
	$\mu_1 + \mu_2 = 3$	†	28	26	24	23	21	20	20	18	18	16	14

be seen that increasing the number of smoother iterations improves the convergence for all the test examples but the matrix FS5414. For this latter matrix, the convergence using M_{Mul} is worse than using simply M_1 . This poor numerical behavior is probably due to the fact that the preconditioned system $M_1 A$ has eigenvalues close to 2. The even number of smoothing iterations moves those eigenvalues near zero, which dramatically affects the numerical behavior of restarted GMRES. Full GMRES succeeds to converge but still using 2 steps of smoothing gives a larger number of iterations than just 1 step. When we move to 3 steps we reduce the number of iterations compared to 2 but also compared to 1 step of smoothing.

Even though no numerical experiments are reported to illustrate this phenomenon, we observed that only the sum $\mu_1 + \mu_2$ plays a role for the convergence of M_{Mul} wherever these smoothing steps are used only as presmoothing, postsmoothing, or split between the pre- and postsmoothing steps. Furthermore, we observe that the relative improvement of using more smoothing steps tends to be larger for a small dimension of the coarse space.

3.1.3. Effect of the relaxation parameter ω . In Table 3 we display the number of iterations required by CG, and by restarted and full GMRES on a set of matrices when the relaxation parameter ω is varied. For SPD matrices, when $\mu_1 + \mu_2$ is odd, no damping is required for the smoother to ensure that the preconditioner is SPD, as illustrated for the matrices 1138BUS and BCSSTK27. It can be seen that $\omega = 1.0$ leads to worse iteration counts than the incomplete factorization alone, i.e., the two space cycles slow down the convergence of CG. This might be due to the fact

TABLE 3
Number of iterations with M_{Add} when the relaxation parameter is varied.

1138BUS: $t = 4 \cdot 10^{-1}$; $\mu_1 + \mu_2 = 3$													
		Dimension of the coarse space											
	ω	M_1	0	1	2	3	4	5	6	7	8	9	10
CG	1.0	295	502	361	273	224	181	174	156	156	146	132	132
	$2\lambda_{\max}^{-1}(M_1 A)$	295	221	161	120	97	81	73	69	69	62	58	58
	$\frac{3}{2}\lambda_{\max}^{-1}(M_1 A)$	295	232	166	128	104	86	81	78	77	66	61	58
	$\lambda_{\max}^{-1}(M_1 A)$	295	251	185	137	111	91	83	79	77	74	69	66
BCSSTK27: $t = 1 \cdot 10^{-1}$; $\mu_1 + \mu_2 = 2$													
		Dimension of the coarse space											
	ω	M_1	0	1	2	3	4	5	6	7	8	9	10
CG	$\frac{3}{2}\lambda_{\max}^{-1}(M_1 A)$	191	139	122	107	94	93	85	75	68	67	67	67
	$\lambda_{\max}^{-1}(M_1 A)$	191	159	141	123	108	106	96	85	77	76	76	76
$\mu_1 + \mu_2 = 3$													
CG	1.0	191	1011	896	810	702	688	621	563	505	503	503	501
	$2\lambda_{\max}^{-1}(M_1 A)$	191	110	98	85	74	73	65	59	53	53	53	53
	$\frac{3}{2}\lambda_{\max}^{-1}(M_1 A)$	191	119	105	91	80	79	71	63	56	56	56	56
	$\lambda_{\max}^{-1}(M_1 A)$	191	138	121	106	93	91	82	74	67	67	67	67
FS-541-4: $t = 8 \cdot 10^{-1}$; $\mu_1 + \mu_2 = 2$													
		Dimension of the coarse space											
	ω	M_1	0	1	2	3	4	5	6	7	8	9	10
GMRES(100)	1.0	126	†	†	†	†	†	†	†	†	†	†	†
	$\frac{3}{2}\lambda_{\max}^{-1}(M_1 A)$	126	80	80	81	80	80	80	70	70	70	69	69
	$\lambda_{\max}^{-1}(M_1 A)$	126	90	88	88	87	87	87	80	80	80	75	75
$\mu_1 + \mu_2 = 3$													
GMRES(∞)	1.0	106	87	86	86	84	84	84	72	72	72	72	72
	$\frac{3}{2}\lambda_{\max}^{-1}(M_1 A)$	106	78	77	77	77	77	77	71	71	71	70	69
	$\lambda_{\max}^{-1}(M_1 A)$	106	67	67	67	65	65	67	59	59	59	58	58
PORES3: $t = 1 \cdot 10^{-1}$; $\mu_1 + \mu_2 = 3$													
		Dimension of the coarse space											
	ω	M_1	0	1	2	3	4	5	6	7	8	9	10
GMRES(10)	1.0	266	120	33	24	19	16	14	13	10	10	9	9
	$\frac{3}{2}\lambda_{\max}^{-1}(M_1 A)$	266	100	28	19	17	13	11	10	9	8	7	7
	$\lambda_{\max}^{-1}(M_1 A)$	266	80	32	25	19	16	14	12	10	9	9	8

that when the number of smoothing steps is increased, they tend to spread the right part of the spectrum if the largest eigenvalues are bigger than two. For this reason, using a damping parameter ensures that the smoother is a contraction that better clusters the right part of the spectrum around one. Nevertheless this contraction may not have a positive effect. For instance, if there were a cluster beyond two, it would be spread by the smoothing iterations, which would possibly penalize the convergence of CG. On the other hand an isolated large eigenvalue would not affect CG, but a scaling to move it below two might shrink the complete spectrum and create clusters near the origin, which might have a negative affect on CG convergence.

For unsymmetric problems, the numerical experiments reveal that the damping also plays a role. Often using a damping parameter improves the convergence of GMRES, especially when the number of smoothing steps is odd. However, no general trends on the suitable choice of this parameter have been revealed by the experiments. Its role is clear in the framework of the stationary iterative method, but a better understanding of the convergence of unsymmetric Krylov solver would be necessary to better understand the influence of this parameter.

3.1.4. Sensitivity to the accuracy of the eigencomputation. As mentioned in the previous section, the eigenvalue calculation is performed in a pre-processing phase. In order to investigate the sensitivity of our algorithm to the eigencomputation accuracy we compute the eigenpairs of a slightly perturbed matrix, $(M_1A + E)$, where E is a random matrix that is scaled so that $\frac{\|E\|}{\|M_1A\|} = \eta$. We use these computed eigenvectors to build our preconditioners and compute the backward error of these vectors as if they were eigenvectors of M_1A . By varying η , we can monitor the level of the backward error that becomes comparable for each eigenvector. In Table 4, we give the number of iterations of the Krylov solvers when the backward error of the computed eigenvectors is varied. As there is one backward error per eigenvector, we give the average of them in the table. It can be seen that, in many cases, there is no need for very high accuracy in the computation of the eigenvectors. On the 685BUS matrix $\eta = 10^{-4}$ ($\eta = 10^{-3}$ for HOR131) gives similar results to $\eta = 10^{-14}$. The numerical behavior of the preconditioner slightly deteriorates when η grows on the GRE1107 matrix, which is probably due to some ill-conditioning of the eigenvectors. Furthermore, it seems that when the number of smoothing steps is increased the accuracy of the eigencalculation has a smaller impact on the numerical efficiency of the preconditioner.

Finally we mention that a sensitivity analysis of some spectral preconditioners is performed in [15] in the SPD cases. For the studied preconditioners, it is shown that their quality depends in particular on the magnitude and the clustering of the eigenvalues that are involved in their definition (i.e., the part of the spectrum they attempt to move). Similarly for the preconditioners studied here, it can be expected that if small or clustered eigenvalues are targeted, a good accuracy should be requested when computing them.

3.2. Implementation in applications. In the previous sections we have illustrated the possible numerical benefit of using a few smoothing iterations to improve the convergence of the preconditioned Krylov solver. In some situations it is the only way to get convergence and the benefit is clear. In other cases it improves the convergence but we might have some questions:

1. Is there a final benefit in terms of computational time because of the extra cost introduced by the calculation of the residual at each iteration of the smoother?
2. Is there a way to alleviate this residual calculation cost?

The answer to the first question is very problem- and machine-dependent. On parallel distributed computers, the dot product calculation in CG and the orthogonalization process in GMRES are the main bottlenecks for performance; reducing the number of iterations is an obvious way to alleviate this cost. When the preconditioner enables a significant reduction of iterations of the Krylov solvers, the extra cost of the matrix-vector products (that usually involves only neighbor-to-neighbor communication) can eventually be compensated for by the decrease of the cost of the dot products (that involves global communications).

One way to reduce this extra cost is to implement a cheaper but approximated matrix-vector product. Such a possibility exists, for instance, when the fast multipole techniques are implemented to compute the matrix-vector product in electromagnetics application. Another example arises in nonoverlapping domain decomposition and is further investigated in the next section.

TABLE 4
Sensitivity of M_{Mul} efficiency versus the accuracy of the eigencomputation.

685BUS: IC($4 \cdot 10^{-1}$); CG; $\mu_1 + \mu_2 = 1$												
Backward error	Dimension of the small dimensional correction space											
	M_1	0	1	2	3	4	5	6	7	8	9	10
$\approx 1 \cdot 10^{-15}$	123	122	83	83	70	57	51	43	40	36	36	34
$\approx 1 \cdot 10^{-6}$	123	123	86	86	72	57	53	43	41	37	37	35
$\approx 1 \cdot 10^{-4}$	123	122	91	90	76	61	57	46	43	39	38	36
$\approx 1 \cdot 10^{-3}$	123	122	127	127	108	94	81	67	59	49	49	44
$\approx 1 \cdot 10^{-2}$	123	122	134	130	290	298	261	257	250	207	210	209
$\mu_1 + \mu_2 = 3$												
$\approx 1 \cdot 10^{-15}$	123	60	43	44	35	28	28	22	20	19	19	17
$\approx 1 \cdot 10^{-6}$	123	60	46	46	38	32	29	24	22	19	18	18
$\approx 1 \cdot 10^{-4}$	123	60	49	49	41	34	25	25	24	21	20	19
$\approx 1 \cdot 10^{-3}$	123	60	64	59	49	48	39	30	30	24	24	22
$\approx 1 \cdot 10^{-2}$	123	60	81	74	76	75	79	63	71	68	67	65
HOR131: ILU($2 \cdot 10^{-1}$); GMRES(30); $\mu_1 + \mu_2 = 1$												
Backward error	Dimension of the small dimensional correction space											
	M_1	0	1	2	3	4	5	6	7	8	9	10
$\approx 1 \cdot 10^{-14}$	93	93	63	57	55	53	51	47	45	39	39	36
$\approx 1 \cdot 10^{-4}$	93	93	63	57	54	52	52	47	45	40	39	35
$\approx 1 \cdot 10^{-3}$	93	93	57	58	58	54	54	48	47	40	41	37
$\approx 1 \cdot 10^{-2}$	93	93	106	114	134	114	150	103	139	137	143	149
$\mu_1 + \mu_2 = 3$												
$\approx 1 \cdot 10^{-14}$	93	56	37	38	32	32	32	26	25	23	22	21
$\approx 1 \cdot 10^{-4}$	93	56	37	38	35	32	32	28	24	22	23	21
$\approx 1 \cdot 10^{-3}$	93	56	36	35	32	31	33	27	27	24	24	21
$\approx 1 \cdot 10^{-2}$	93	56	53	57	68	63	73	53	79	59	58	82
GRE1107: ILU($1 \cdot 10^{-2}$); GMRES(30); $\mu_1 + \mu_2 = 1$												
Backward error	Dimension of the small dimensional correction space											
	M_1	0	1	2	3	4	5	6	7	8	9	10
$\approx 1 \cdot 10^{-14}$	†	†	80	76	37	34	32	31	27	26	24	23
$\approx 1 \cdot 10^{-8}$	†	†	95	80	40	36	33	34	31	30	29	28
$\approx 1 \cdot 10^{-7}$	†	†	117	105	77	40	38	38	34	33	33	25
$\approx 1 \cdot 10^{-6}$	†	†	440	80	79	78	77	67	40	74	39	38
$\mu_1 + \mu_2 = 3$												
$\approx 1 \cdot 10^{-14}$	†	28	26	25	22	21	19	18	17	16	15	14
$\approx 1 \cdot 10^{-8}$	†	28	27	26	24	23	21	21	19	19	18	18
$\approx 1 \cdot 10^{-7}$	†	28	27	27	26	24	23	24	21	21	21	20
$\approx 1 \cdot 10^{-6}$	†	28	29	27	27	27	26	25	26	27	25	25

3.2.1. A case study in semiconductor device simulation. The numerical simulation of two-dimensional semiconductor devices is extremely demanding in terms of computational time because it involves complex embedded numerical schemes. At the kernel of these schemes is the solution of very ill-conditioned large linear systems. Such problems are challenging because they are large, very ill-conditioned, and extremely badly scaled. In that respect, robust and efficient linear solvers should be selected in order to reduce the elapsed time required to successfully perform a complete simulation. In this section we present some numerical experiments obtained by

TABLE 5

Experiments with M_{Mul} on SPD systems arising in semiconductor device modeling when the density of \tilde{S} is varied; $\omega = \frac{3}{2}\lambda_{\max}^{-1}(M_1A)$.

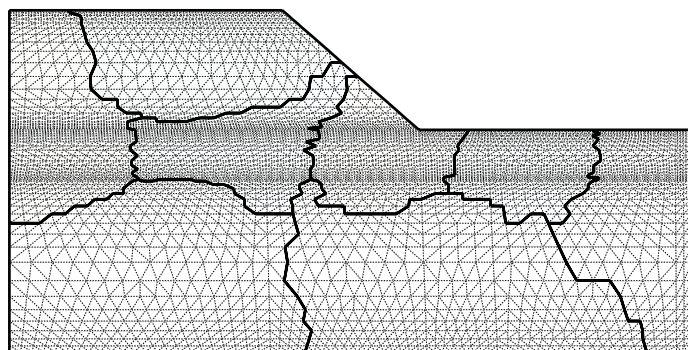
Density	$\mu_1 + \mu_2 = 2$										
	Dimension of the coarse space										
	M_1	1	2	3	4	5	6	7	8	9	10
29%	52	34	39	38	41	38	37	37	37	37	36
42%	52	30	27	25	22	20	19	18	18	19	18
49%	52	28	25	24	21	19	18	17	17	17	17
58%	52	28	25	24	21	19	18	17	17	17	17
100%	52	28	25	24	21	19	18	17	17	17	17

using the spectral two-level preconditioners for the solution of the Schur complement systems resulting from a nonoverlapping domain decomposition approach. For that example the smoother is defined by an additive Schwarz preconditioner for the Schur complement system; we refer the reader to [16] for more details on this application.

In this implementation of the iterative substructuring approach the Schur complement matrix that is sparse with dense blocks is computed explicitly. In order to reduce the computational cost of the residual calculation involved in each step of the smoother, we consider a sparsified Schur complement to perform this operation. This sparse approximation of the Schur complement is obtained by dropping all the entries that are smaller than a prescribed threshold. Even though we have not considered other variants to get a cheap approximation, we mention that other possibilities exist. For instance, if the Schur complement were not explicitly formed, we might have considered some approximations computed using incomplete factorizations for the local subproblems or using a probing technique [9].

In Table 5 we report numerical experiments observed on one of the SPD linear systems that has to be solved in our semiconductor device simulation. It corresponds to a partitioning of the domain into eight subdomains, as shown in Figure 3. The complete domain is discretized by 155,000 degrees of freedom (dof), and the size of the interface between the subdomains (i.e., the size of the associated Schur complement matrix) is 1607. In this table, we vary the sparsity of \tilde{S} , the sparse approximation of the Schur complement matrix, and report the number of CG iterations required to obtain a reduction of the normalized unpreconditioned residual by 10^{-11} . This very small threshold for the stopping criterion of the linear solver was required to ensure the convergence of the nonlinear scheme. It can be seen that when the size of the coarse space is varied, more than half of the entries of S can be dropped for the implementation of the smoother without affecting the numerical behavior of M_{Mul} . This enables us to save a significant amount of floating-point operations involved in the smoother and consequently to reduce its cost substantially. Dropping 70% of the entries doubles the number of iterations while it divides by three the cost of applying the preconditioner. Depending on the target computer this latter situation might possibly lead to saving time.

3.2.2. A case study in electromagnetism applications. Electromagnetic scattering problems give rise to linear systems that are challenging to solve by iterative methods. For three-dimensional problems the boundary integral formulation of Maxwell's equations is often selected because of its nice approximation properties. This approach gives rise to dense complex systems. In recent years, the introduction of fast methods with reduced computational complexity and storage requirement has

FIG. 3. *Mesh partitioning of the semiconductor discretization.*

attracted an increasing interest in the use of preconditioned Krylov methods for the simulation of real-life electromagnetic problems. In this section, we consider the surface integral formulation of Maxwell's equations modeled via the electric-field integral equation (EFIE), which is the most general but the most difficult formulation to solve for iterative solvers. Thus preconditioning is crucial and approximate inverse methods based on Frobenius-norm minimization have proved to be among the most effective preconditioners for solving these systems efficiently. We refer the reader to [7] and the references therein for a more detailed presentation of this preconditioning technique on this class of problems.

The Frobenius-norm minimization preconditioner M_{FROB} is very effective in clustering most of the eigenvalues near one, but tends to leave a few isolated eigenvalues close to zero. The presence of these very small eigenvalues can slow down the convergence of iterative solvers especially on large problems, where the memory constraints prevent the use of large restarts in GMRES, which is the best-suited solver for those problems. In this section, we apply the two-space spectral preconditioner on top of the Frobenius-norm minimization method. The preconditioner M_{FROB} is actually used to define a stationary iterative scheme implemented as a smoother. In the numerical experiments, the initial guess is the zero vector, we consider a right preconditioned GMRES method, and the threshold for the stopping criterion is set to 10^{-3} on the normwise backward error $\frac{\|r\|}{\|b\|}$, where r denotes the residual and b the right-hand side of the linear system. This tolerance is accurate for engineering purposes, as it enables the correct reconstruction of the radar cross-section of the objects. The parallel runs have been performed in single precision complex arithmetic on sixteen processors of an HP-Compaq Alpha server, a cluster of symmetric multiprocessors. We report results on two industrial problems, namely, a cobra (Figure 4(a)) discretized with 60,695 dof and an almond (Figure 4(b)) discretized with 104,793 dof. We are in the suited situation, where the first-level preconditioner M_{FROB} does a good job in clustering most of the eigenvalues near one and leaves only a few outliers close to the origin. Consequently, the eigenvectors are computed in forward mode by ARPACK in a preprocessing phase.

For the matrix-vector products, we use the fast multipole method (FMM) [18], which performs fast matrix-vector products in $O(n \log n)$ arithmetic operations. More precisely, a highly accurate FMM is used for the standard matrix-vector product operation within GMRES, and a less accurate FMM is used for the extra matrix-vector products involved in the smoother of the preconditioning operation. The price

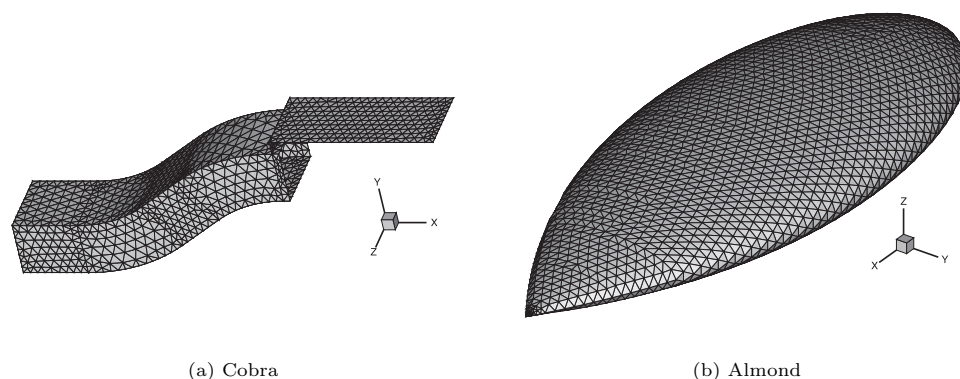


FIG. 4. Mesh associated with test examples.

to pay for the accuracy is mainly computational time; high accuracy also means more computational time.

In Table 6 we report on the number of iterations and the elapsed time for different sizes of the coarse space and an increasing number of smoothing steps. For the two test examples we show results with restarted and full GMRES. For these experiments we set up $\omega^{-1} = \frac{2}{3}\lambda_{\max}(M_{FROB}A)$, which seems to be an overall good choice. On these problems M_{Add} and M_{Mul} give the same number of iterations. For this reason we report only on the M_{Add} preconditioner, which is about 33% faster than M_{Mul} as it requires fewer matrix-vector operations, as indicated in section 2.4. For instance, on the almond test problem using 3 smoothing steps and 50 eigenvectors, both converge in 87 iterations with GMRES(10), M_{Mul} takes 10 minutes, and M_{Add} only 6 minutes.

In this table we also display the number of iterations and the elapsed time of GMRES with only M_{FROB} . It can be seen that the use of M_{Add} is always beneficial both from a number iterations viewpoint and from a computational time viewpoint. The gain in time varies from 14 to infinity with GMRES(10) and between 2 and 3 with GMRES(∞).

From a numerical point of view we observe in these examples the same behavior as before. That is, the larger the coarse space, the better the preconditioner; the number of GMRES iterations decreases when the number of smoothing steps is increased. Furthermore, the gain is larger if restarted GMRES is considered than if full GMRES is used as solver. In particular on the almond with GMRES(10) and less than 50 eigenvectors, the only way to get convergence is to perform a few steps of smoothing; with 50 eigenvectors the gain introduced by the smoothing iterations is still tremendous (i.e., larger than 21). On the Cobra problem, using 15 eigenvectors the gain is far larger than 2 with GMRES(10), and close to 2 for full GMRES.

Not only is the number of iterations significantly reduced, but the solution time is as well. On the almond problem, using 50 eigenvectors and GMRES(10) we gain a factor of 12 in elapsed time when we increase the number of smoothing steps from 1 to 3. We mention that on large electromagnetic problems (of size larger than 0.5 million unknowns) the use of small restarts is recommended in order to save the heavy cost of reorthogonalization and reduce the final solution cost. The choice of a small restart is also dictated by memory constraints [7]. With full GMRES, the number of iterations is significantly decreased but the total solution cost is likely to only slightly decrease

TABLE 6
Experiments with M_{Add} on the electromagnetics problems.

Cobra problem			
With M_{FROB} GMRES(10) 2719 iterations (1 h 10 min)			
GMRES(∞) 378 iterations (18 min)			
$\mu_1 + \mu_2 = 1$			
Dimension of the coarse space			
	5	10	15
GMRES(10)	1458 (42 min)	594 (12 min)	517 (11 min)
GMRES(∞)	262 (9 min)	216 (7 min)	188 (6 min)
$\mu_1 + \mu_2 = 2$			
Dimension of the coarse space			
	5	10	15
GMRES(10)	471 (18 min)	209 (7 min)	201 (6 min)
GMRES(∞)	161 (8 min)	132 (6 min)	115 (5 min)
$\mu_1 + \mu_2 = 3$			
Dimension of the coarse space			
	5	10	15
GMRES(10)	281 (12 min)	132 (6 min)	124 (5 min)
GMRES(∞)	120 (7 min)	98 (6 min)	85 (5 min)

Almond problem			
With M_{FROB} GMRES(10) +3000 iterations			
GMRES(∞) 242 iterations (14 min)			
$\mu_1 + \mu_2 = 1$			
Dimension of the coarse space			
	10	30	50
GMRES(10)	+3000	+3000	1867 (1 h 12 min)
GMRES(∞)	229 (13 min)	157 (8 min)	132 (6 min)
$\mu_1 + \mu_2 = 2$			
Dimension of the coarse space			
	10	30	50
GMRES(10)	552 (29 min)	245 (14 min)	176 (9 min)
GMRES(∞)	134 (9 min)	92 (6 min)	77 (6 min)
$\mu_1 + \mu_2 = 3$			
Dimension of the coarse space			
	10	30	50
GMRES(10)	216 (16 min)	116 (9 min)	87 (6 min)
GMRES(∞)	97 (9 min)	66 (6 min)	56 (6 min)

when the number of smoothing steps is large as the preconditioner is expensive to apply. The optimal selection of the size of the coarse space and of the number of smoothing steps remains an open question, and the choice mainly depends on the clustering properties of the initial preconditioner. In terms of computational cost the coarse grid correction and the smoothing mechanism are complementary components that have to be suitably combined. On the cobra problem, for instance, using 3 smoothing steps and 5 eigenvectors we obtain a better convergence rate but similar computational time than when using 1 smoothing step but 15 eigenvectors; on the almond problem, the solution cost of GMRES(10) with 3 smoothing steps and 30 eigenvectors is similar to the time with 2 smoothing steps but 50 eigenvectors. Enabling a reduction in the number of eigenvectors by using more smoothing steps is a desirable feature in contexts where computing many eigenvalues can become very expensive, which is, for instance, the case in this application for very large problems.

To illustrate the cost related to the eigenvector calculation, we mention that the computation of 30 eigenvectors in the almond problem requires 1100 matrix-vector products, that is, about 2 hours. In the cobra example, 500 matrix-vector products

are needed to calculate 15 eigenvectors, which corresponds to 1 hour. The extra cost associated with this preprocessing is quickly amortized as many right-hand sides have usually to be solved to compute the so-called radar cross-section. In the cobra example, the eigencalculation is amortized if more than 2 right-hand sides are solved using GMRES(10) or more than 5 if full GMRES can be afforded. For the almond, the gain is unbounded if GMRES(10) is used and 15 right-hand sides should be solved to amortize the eigencalculation if full GMRES can be afforded. To conclude, we mention that numerical experiments are reported in [10] on the same test problems with the preconditioner presented in [6]. The observed behavior is similar to those reported here when $\mu_1 + \mu_2 = 1$, which again illustrates the benefits of the new approach when only a few eigenpairs are available.

4. Concluding remarks. In this work, we exploit some ideas implemented in the multigrid techniques for the solution of PDEs to derive new additive and multiplicative spectral two-level preconditioners for the solution of general linear systems. We propose a scheme that enables us to improve a given preconditioner that leaves only few eigenvalues close to zero. We study the spectrum of the new preconditioned matrix associated with these schemes. The effectiveness and robustness of these preconditioners is mainly due to their ability to shift to one a selected set of eigenvalues and to cluster near one most of the others. We illustrate some features of these techniques in small examples from the Matrix Market and illustrate their attractive numerical behavior on two challenging real-world applications. On electromagnetism industrial problems, we show that the preconditioner enables us to save a significant amount of time for the solution of large problems on parallel platforms.

Acknowledgments. The authors would like to thank Ray Tuminaro for the stimulating discussions he had with the second author when he visited him at Sandia. The authors are grateful to Emeric Martin for his assistance in implementing the preconditioner in the electromagnetism code. Finally, we would like to thank the anonymous referees for their valuable remarks, questions, and comments that enabled us to improve this paper.

REFERENCES

- [1] J. BAGLAMA, D. CALVETTI, G. H. GOLUB, AND L. REICHEL, *Adaptively preconditioned GMRES algorithms*, SIAM J. Sci. Comput., 20 (1998), pp. 243–269.
- [2] P. BASTIAN, W. HACKBUSH, AND G. WITTUM, *Additive and multiplicative multigrid: A comparison*, Computing, 60 (1998), pp. 345–364.
- [3] M. BENZI, C. D. MEYER, AND M. TUMA, *A sparse approximate inverse preconditioner for the conjugate gradient method*, SIAM J. Sci. Comput., 17 (1995), pp. 1135–1149.
- [4] R. BOISVERT, R. POZO, K. REMINGTON, R. BARRETT, AND J. DONGARRA, *Matrix Market: A web resource for test matrix collections*, in The Quality of Numerical Software: Assessment and Enhancement, R. Boisvert, ed., Chapman and Hall, London, 1997, pp. 125–137.
- [5] J. H. BRAMBLE, J. E. PASCIAK, AND A. H. SCHATZ, *The construction of preconditioners for elliptic problems by substructuring I*, Math. Comp., 47 (1986), pp. 103–134.
- [6] B. CARPENTIERI, I. S. DUFF, AND L. GIRAUD, *A class of spectral two-level preconditioners*, SIAM J. Sci. Comput., 25 (2003), pp. 749–765.
- [7] B. CARPENTIERI, I. S. DUFF, L. GIRAUD, AND G. SYLVAND, *Combining fast multipole techniques and an approximate inverse preconditioner for large electromagnetism calculations*, SIAM J. Sci. Comput., 27 (2005), pp. 774–792.
- [8] B. CARPENTIERI, L. GIRAUD, AND S. GRATTON, *Additive and Multiplicative Two-Level Spectral Preconditioning for General Linear Systems*, Technical report TR/PA/04/38, CERFACS, Toulouse, France, 2004.
- [9] T. F. C. CHAN AND T. P. MATHEW, *The interface probing technique in domain decomposition*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 212–238.

- [10] I. S. DUFF, L. GIRAUD, J. LANGOU, AND E. MARTIN, *Using spectral low rank preconditioners for large electromagnetic calculations*, Internat. J. Numer. Methods Engrg., 62 (2005), pp. 416–434.
- [11] J. ERHEL, K. BURRAGE, AND B. POHL, *Restarted GMRES preconditioned by deflation*, J. Comput. Appl. Math., 69 (1996), pp. 303–318.
- [12] M. FISHER, *Minimization algorithms for variational data assimilation*, in Proceedings of the ECMWF Seminar “Recent Developments in Numerical Methods for Atmospheric Modelling,” 1998, pp. 364–385.
- [13] L. FOURNIER AND S. LANTERI, *Multiplicative and additive parallel multigrid algorithms for the acceleration of compressible flow computations on unstructured meshes*, Appl. Numer. Math., 36 (2001), pp. 401–426.
- [14] J. FRANK AND C. VUIK, *On the construction of deflation-based preconditioners*, SIAM J. Sci. Comput., 23 (2001), pp. 442–462.
- [15] L. GIRAUD AND S. GRATTON, *On the sensitivity of some spectral preconditioners*, SIAM J. Matrix Anal. Appl., 27 (2006), pp. 1089–1105.
- [16] L. GIRAUD, A. MARROCCO, AND J.-C. RIOUAL, *Iterative versus direct parallel substructuring methods in semiconductor device modelling*, Numer. Linear Algebra Appl., 12 (2005), pp. 33–53.
- [17] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 3rd ed., Johns Hopkins Studies in the Mathematical Sciences, The Johns Hopkins University Press, Baltimore, MD, 1996.
- [18] L. GREENGARD AND V. ROKHLIN, *A fast algorithm for particle simulations*, J. Comput. Phys., 73 (1987), pp. 325–348.
- [19] W. HACKBUSCH, *Multigrid Methods and Applications*, Springer-Verlag, Berlin, 1985.
- [20] S. A. KHARCHENKO AND A. YU. YEREMIN, *Eigenvalue translation based preconditioners for the GMRES(k) method*, Numer. Linear Algebra Appl., 2 (1995), pp. 51–77.
- [21] L. YU. KOLOTLINA AND A. YU. YEREMIN, *Factorized sparse approximate inverse preconditionings I. Theory*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 45–58.
- [22] C. LE CALVEZ AND B. MOLINA, *Implicitly restarted and deflated GMRES*, Numer. Algorithms, 21 (1999), pp. 261–285.
- [23] R. B. LEHOUCQ, D. C. SORESENSEN, AND C. YANG, *ARPACK Users’ Guide: Solution of Large-Scale Problems with Implicitly Restarted Arnoldi Methods*, SIAM, Philadelphia, 1998.
- [24] J. A. MEIJERINK AND H. A. VAN DER VORST, *An iterative solution method for linear systems of which the coefficient matrix is a symmetric M -matrix*, Math. Comp., 31 (1977), pp. 148–162.
- [25] R. B. MORGAN, *GMRES with deflated restarting*, SIAM J. Sci. Comput., 24 (2002), pp. 20–37.
- [26] R. A. NABBen AND C. VUIK, *A comparison of deflation and coarse grid correction applied to porous media flow*, SIAM J. Numer. Anal., 42 (2004), pp. 1631–1647.
- [27] R. A. NICOLAIDES, *Deflation of conjugate gradients with applications to boundary value problems*, SIAM J. Numer. Anal., 24 (1987), pp. 355–365.
- [28] A. PADIY, O. AXELSSON, AND B. POLMAN, *Generalized augmented matrix preconditioning approach and its application to iterative solution of ill-conditioned algebraic systems*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 793–818.
- [29] Y. SAAD, *Projection and deflation methods for partial pole assignment in linear state feedback*, IEEE Trans. Automat. Control, 33 (1988), pp. 290–297.
- [30] Y. SAAD, *Analysis of augmented Krylov subspace techniques*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 435–449.
- [31] Y. SAAD, *ILUT: A dual threshold incomplete LU factorization*, Numer. Linear Algebra Appl., 1 (1994), pp. 387–402.
- [32] Y. SAAD, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, 2003.
- [33] Y. SAAD AND M. H. SCHULTZ, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.
- [34] R. S. TUMINARO, *A highly parallel multigrid-like method for the solution of the Euler equations*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 88–100.
- [35] H. A. VAN DER VORST, *Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 631–644.
- [36] H. WAISMAN, J. FISH, R. S. TUMINARO, AND J. SHADID, *The generalized global basis method*, Internat. J. Numer. Methods Engrg., 61 (2004), pp. 1243–1269.